



**UNITED STATES DEPARTMENT OF COMMERCE**  
**Patent and Trademark Office**

Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
-----------------	-------------	----------------------	---------------------

09/282,907 03/31/99 CHAO

C AT9-98-441

EXAMINER

TM02/0703

WINSTEAD SECHREST & MINICK  
100 CONGRESS AVENUE  
SUITE 800  
AUSTIN TX 78701

FLEURANTIN, J

ART UNIT

PAPER NUMBER

2172

DATE MAILED:

07/03/01

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

# Office Action Summary

Application No.  
09/282,907

Applicant(s)  
Ching-Yun Chao et al.

Examiner  
Jean Bolte Fleurantin

Art Unit  
2172



-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE three MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_\_
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above, claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claims \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are objected to by the Examiner.
- 11) ☐ The proposed drawing correction filed on \_\_\_\_\_ is: a) ☐ approved b) ☐ disapproved.
- 12) ☐ The oath or declaration is objected to by the Examiner.

## Priority under 35 U.S.C. § 119

- 13) ☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
- a) ☐ All b) ☐ Some\* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \*See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

## Attachment(s)

- 15) ☒ Notice of References Cited (PTO-892) 18) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 16) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948) 19) ☐ Notice of Informal Patent Application (PTO-152)
- 17) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s). 4 20) ☐ Other: \_\_\_\_\_

Art Unit: 2172

### DETAILED ACTION

1. Claim 1-30 are presented for examination.

#### *Disclosure Objections*

2. The disclosure is objected to because of the first paragraph on page 1 should be updated with the current status of the US Application, Serial Number, a filing data, and the issue date.

Appropriate correction is required.

#### *Claim Rejections - 35 U.S.C. § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-30 are rejected under 35 U.S. C. 103 (a) as being unpatentable over San Andres et al. (US Pat. No. 5,956,489).

As per claim 1, San Andres substantially teaches a method for maintaining a consistent set of replicas of a database within a computer cluster (thus, enabling each application server to maintain a replicated copy of service content data, which is readable as maintaining a consistent set of replicas of a database within a computer cluster) (see abstract, lines 5-6) as claimed comprises the steps of each node in the computer cluster receiving a database update request (thus, once the content of the application server is brought up to date the application server is

Art Unit: 2172

placed in a state which allows it to receive client request, which is readable as each node in the computer cluster receiving a database update request)(see col. 3, lines 8-11, 26-30);

detecting an out-of-sync condition as a result of a different functional outcome (thus, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as detecting an out-of-sync condition as a result of a different functional outcome) (see col. 2, lines 55-60). But, San Andres does not explicitly indicate the step of each node in the computer cluster voting based on a functional outcome of the database update request. However, implicitly San Andres shows the step of each time an update transaction is dispatched by the arbiter the arbiter monitors the outcome of the transaction on each server by checking the status codes returned by the servers, when one server of the service group processes the dispatched transaction differently than the other servers the arbiter uses a voting scheme to decide which server or servers are to be taken off line service group, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is readable as each node in the computer cluster voting based on a functional outcome of the database update request (see col. 19, lines 43-55). It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teaching of San Andres with the step of each node

Art Unit: 2172

in the computer cluster voting based on a functional outcome of the database update request.

This modification would allow the teachings of San Andres to provide access to identical data and so that the on line service appears the same to all end users (col. 1, lines 61-61).

As per claim 2, San Andres substantially teaches a method as claimed, wherein the out-of-sync condition is an error (thus, the arbiter uses a majority rules voting scheme under the majority rules scheme, if the majority number servers of the service group report a different outcome than others servers the majority servers are treated as being inconsistent with final outcome and taken off line; which is readable as wherein the out-of-sync condition is an error) (see col. 19, lines 18-55).

As per claim 3, San Andres substantially teaches a method as claimed, further comprises the step of: refreshing the database in response to the detecting step (thus, when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update transaction to the arbiter, which is readable as refreshing the database in response to the detecting) (see col. 3, lines 26-30).

As per claim 4, San Andres substantially teaches a method as claimed, further comprises the step of: resetting cluster membership in response to the detecting step (thus, when an application server of a service group receives a client request that indicates a modification to replicated service content data the server generates an update transaction and sends the update

Art Unit: 2172

transaction to the arbiter, which is readable as resetting cluster membership in response to the detecting (see col. 3, lines 26-30).

As per claim 5, San Andres substantially teaches a method as claimed, further comprises the step of blocking further participation by the node having the out-of-sync condition in response to the detecting step (thus, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as participation by the node having the out-of-sync condition in response to the detecting step (see col. 2, lines 55-60).

As per claim 6, San Andres substantially teaches a method as claimed, further comprises the step of: declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner (thus, servers receive the update transaction from arbiter respond by processing the update transaction and returning a status code that indicates the success or failure, which is equivalent to declaring an end-of-transaction state on update voting completion when the database update is being done in a transactional manner) (see col. 19, lines 25-28).

As per claim 7, San Andres substantially teaches a method as claimed, further comprising the step of backing out an update when update voting does not meet a criteria established for success (thus, servers receive the update transaction from arbiter respond by processing the update transaction and returning a status code that indicates the success or failure, which is

Art Unit: 2172

readable as backing out an update when update voting does not meet a criteria established for success) (see col. 19, lines 25-28).

As per claim 8, San Andres substantially teaches a method as claimed, wherein the criteria established for success is that no more than one node has inconsistent results (see col. 2, lines 55-64).

As per claim 9, in addition to the discussion in claim 1, San Andres teaches a method for maintaining a consistent set of replicas of a database within a computer cluster (see col. 1, lines 60-61) as claimed, comprises the steps of: broadcasting an update to a database shared among a plurality of nodes in the computer cluster (thus, the arbiter records the update transaction in a service group specific transaction log and forwards the transaction for immediate processing to every application server in the group, which is equivalent to broadcasting an update to a database shared among a plurality of nodes in the computer cluster) (see col. 3, lines 30-34);

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster (thus, all duplicated servers of the service group maintain local copies of the service's content data and provide user access to such data, which is readable as applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster) (see col. 9, lines 19-21);

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database (thus, a conflict resolution feature for resolving transaction processing conflicts between application servers, when different

Art Unit: 2172

application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database) (see col. 2, lines 53-60).

As per claims 10, 16, and 26 San Andres substantially teaches a method as claimed, further comprising the step of: voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison (thus, when inconsistent transaction results are reported by different application servers the transaction replication service uses a voting scheme to decide which application servers are to be deemed 'consistent'; which is readable as voting, by any one of the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison) (see abstract, lines 10-14).

As per claims 11, 17, 22, and 27 San Andres substantially teaches a method as claimed, further comprising the step of broadcasting an approval of the update to the database if all of the other nodes vote to approve the update (thus, when a new application server is brought on line previously dispatched update transactions stored in the transaction log are dispatched in sequence to the new server to bring the new server's content data up to date, which is readable as



Art Unit: 2172

broadcasting an approval of the update to the database if all of the other nodes vote to approve the update) (see abstract, lines 16-20).

As per claims 12, 18, 23, and 28 San Andres substantially teaches a method as claimed, further comprises the step of if more than one of the plurality of nodes votes to continue, performing a recovery process (see abstract, lines 11-13).

As per claims 13, 19, 24, and 29 San Andres substantially teaches a method as claimed, wherein the recovery process further comprises the step of if more than a specified number of the nodes voted to continue, backing out the update to the database (see col 20, lines 26-42).

As per claims 14, 20, and 30 San Andres substantially teaches a method as claimed 14. The method as recited in claim 12, wherein the recovery process further comprises the step of: if less than a specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes (see col. 17, lines 10-19).

As per claim 15, in addition to the discussion in claims 1 and 9, San Andres substantially teaches a computer cluster operable for maintaining a consistent set of replicas of a database within the computer cluster as claimed, comprises: a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster (thus, architecture of on line services network in which a preferred embodiment of a transaction replication system and method are employed, which is readable as a group services client operable for broadcasting an update to a database shared among a plurality of nodes in the computer cluster) (see figure 1, col. 3, lines 50-53);

Art Unit: 2172

the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster (thus, a conflict resolution feature for resolving transaction processing conflicts between application servers, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as the plurality of nodes coupled to the computer cluster operable for applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster) (see col. 2, lines 53-60).

As per claim 21, in addition to the discussion in claim 9, San Andres substantially teaches a computer program product adaptable for storage on a computer readable medium, the computer program product operable for maintaining a consistent set of replicas of a database within a computer cluster as claimed, comprises the program steps of voting, by any one if the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison (the transaction replication service uses a voting scheme to decide which application servers are to be deemed 'consistent'; which is readable as voting, by any one if the other nodes in the computer cluster, to continue with update process if a match does not result from the comparison) (see abstract, lines 11-13).

As per claim 25, San Andres substantially teaches a computer program product as claimed, wherein the recovery process further comprises the program step of: if less than a

Art Unit: 2172

specified number of the nodes voted to continue, performing the recovery process on the specified number of the nodes (see col. 17, lines 10-19).

A method for maintaining a consistent set of replicas of a database within a computer cluster, comprising the steps of broadcasting an update to a database shared among a plurality of nodes in the computer cluster (see figure 1, col. 3, lines 50-53);

applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster (thus, a conflict resolution feature for resolving transaction processing conflicts between application servers, when different application servers of a service group process the same update transaction differently the arbiter resolves the conflict by determining the final outcome of the transaction for the service group as whole and taking by any application servers off line that are in conflict with this final outcome, which is readable as applying the update to a local copy of the database at each of the plurality of nodes in the computer cluster) (see col. 2, lines 53-60);

node requesting update broadcasts results of update to all of the other nodes in the computer cluster (see col. 19, lines 43-55);

comparing, by all of the other nodes in the computer cluster, the update results to results of application of the update to the local copy of the database (see cols 31 and 32, lines 64-67 and 1-2);

voting, by all of the other nodes in the computer cluster, to approve update if a match results from the comparison (see abstract, lines 10-14).

Art Unit: 2172

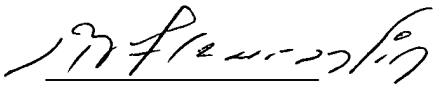
4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Jain et al. US Patent Number 5,737,601 relates to the field of data replication.

*Conclusion*

5. Any inquiry concerning this communication from examiner should be directed to Jean Bolte Fleurantin at (703) 308-6718. The examiner can normally be reached on Monday to Friday from 7:30 A.M. to 6.00 P.M.

If any attempt to reach the examiner by telephone is unsuccessful, the examiner's supervisor, Mrs. KIM VU can be reached at (703) 305-8449. The FAX phone number is (703) 305-9731.


Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone is (703) 305-9600.



Jean Bolte Fleurantin

June 29, 2001

JBf/



KIM VU  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100